

```
1 Buat program menghitung faktorial dari N.
2 Contoh:
3 > (faktorial 5)
4 120
5
6 ?- faktorial(5,X).
7 X=120
8
9 faktorial(0,1).
10 faktorial(N,X) :-
11     A is N-1,
12     faktorial(A,M),
13     X is N*M.
14
15 Buat program untuk menerima tiga bilangan bulat tidak
... negatif a, b, dan c, kemudian menghitung nilai dari
... (a+b)/(c+1)+1.
16 Contoh:
17 > (hitung 6 2 3)
18 3.0
19 ?- hitung(6,2,3,X).
20 X=3
21
22 (define (hitung a b c)
23     (exact -> inexact
24         (+ (/ (+ a b) (+ c 1)) 1)
25     )
26 )
27
28 hitung(A,B,C,X) :-
29     X1 is A+B,
30     X2 is C+1,
31     X is X1/X2+1.
32
33 Buat program menghitung jumlah digit bilangan bulat positif.
34 Contoh:
35 > (digit 234)
36 9
37 ?- digit(234,X).
38 X=9
39
40 (define (digit n)
```

```
41 (if (< n 10) n
42     (+ (modulo n 10) (digit (floor (/ n 10)))))
43 )
44 )
```

```
45
46 digit(N,N) :- N<10.
47 digit(N,X) :-
48     A is N mod 10,
49     B is N//10,
50     digit(B,C),
51     X is A+C.
```

52
53 Buat program menghitung banyaknya elemen list sederhana.

54 Contoh:

```
55 > (panjang '(2 3 4))
56 3
57 ?- panjang([2,3,4],X).
58 X=3
```

```
59
60 (define (panjang x)
61     (if (empty? x) 0
62         (+ 1 (panjang (cdr x)))))
```

```
63
64 panjang([], 0).
65 panjang(_|T, N) :-
66     panjang(T,A),
67     N is 1+A.
```

68
69
70 Buat program mengambil elemen pertama dari list sederhana.

71 Contoh:

```
72 ?- ambil([2,3,4],X).
73 X=2
```

```
74
75 ambil([A|_],X) :- X is A.
76 atau
77 ambil([A|_],A).
```

78
79
80 Buat program mengambil elemen terakhir dari list sederhana.

```
81
82 (define (ambil x)
```

```
83     (if (empty? (cdr x)) (car x)
84         (ambil (cdr x)))
85
86 ambil([A|[]], A).
87 ambil([_|T], X) :- ambil(T,X).
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
```

125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146